Release A Enhancements

Jim Dennison/Dan Trieschman

jdenniso@eos.hitc.com/dtriesch@eos.hitc.com

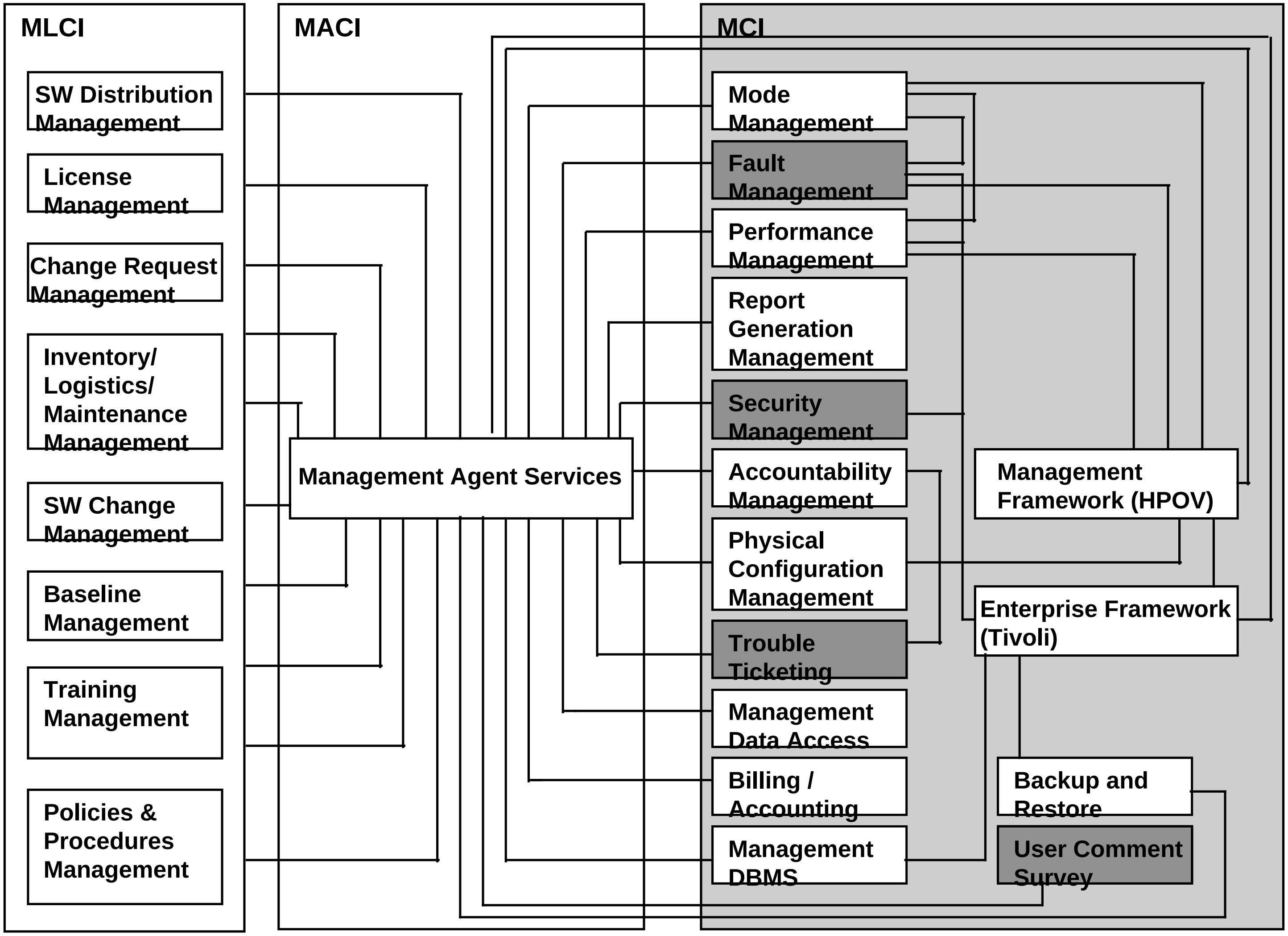
16 April 1996

Overview

Release A Enhancements:

- Addressed in this presentation:
  - Fault Management
  - Trouble Ticketing
  - Security Management
  - User Comment Survey
- Addressed in separate presentations:
  - Accountability Management/Request Tracking
  - Management Agent
  - Management Data Access

MSS Software Architecture Overview


MLCI
SW Distribution Management
License Management
Change Request Management
Inventory/ Logistics/ Maintenance Management
SW Change Management
Baseline Management
Training Management
Policies & Procedures Management
MACI
Management Agent Services
MCI
Mode Management
Fault Management
Performance Management
Report Generation Management
Security Management
Accountability Management
Physical Configuration Management
Trouble Ticketing
Management Data Access
Billing / Accounting
Management DBMS
Management Framework (HPOV)
Enterprise Framework (Tivoli)
Backup and Restore
User Comment Survey

Fault Management Driving Requirements

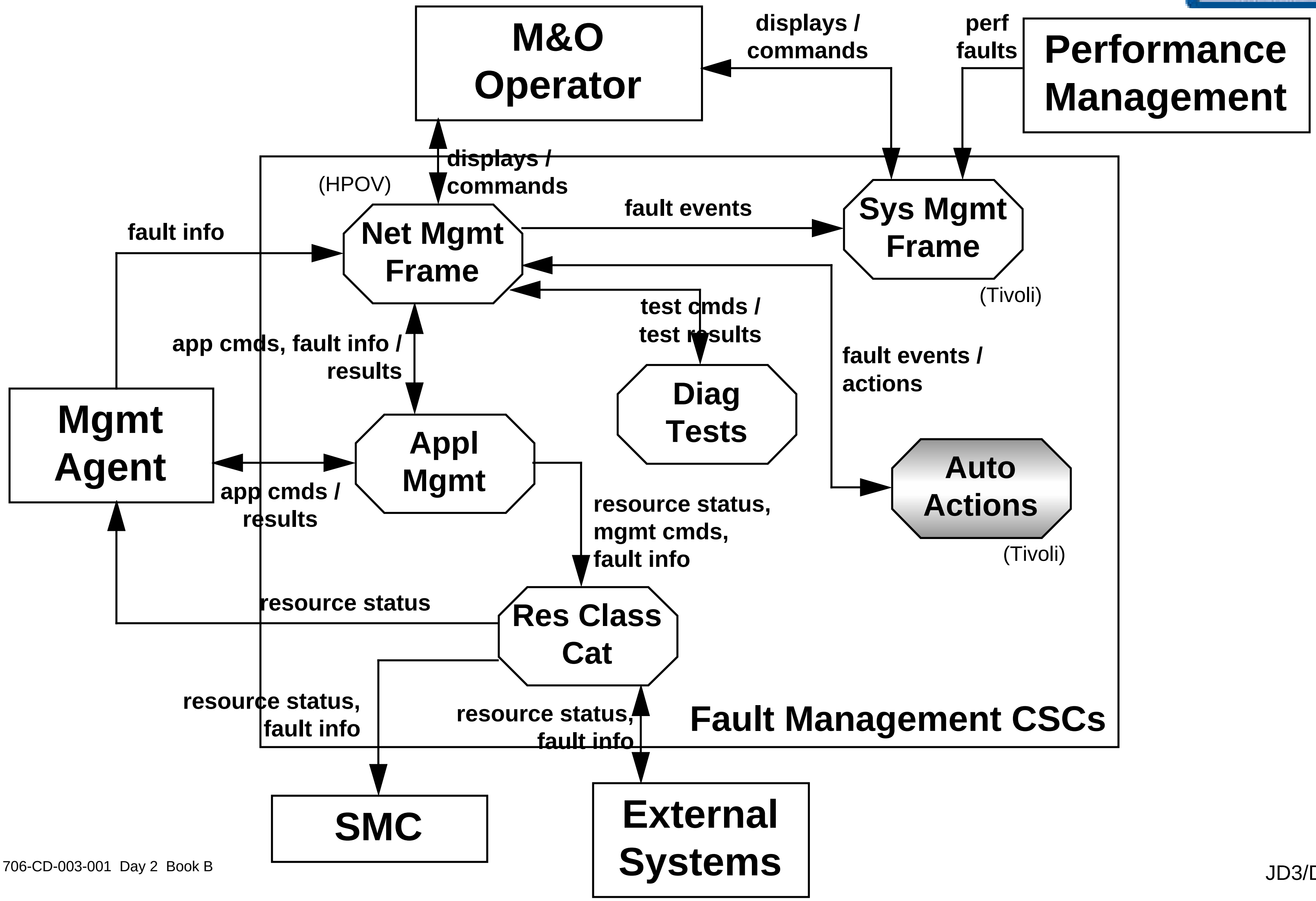
Requirement Summary

- Monitor ECS network and applications for faults
- Provide fault filtering for operator
- Alert operator of significant faults
- Provide operator with control of ECS applications

Changes for Release B

- Provide Fault Correlation

Fault Management Software Architecture



Fault Correlation

Performed by Tivoli SENTRY and ENTERPRISE CONSOLE

- Release B Functionality
- Enterprise Console provides GUI to monitor and flag events.
  - Icons
  - Status/Alerts
  - Log and Browser
- Sentry provides event monitoring and threshold setting at local level.
- COTS Event Adapter provided to receive events from HP Openview
- COTS Log Event Adapter to monitor System and Application logs
- Tool kits provided to develop event Adapters for future enhancements
  - Event Adapters may be supplied by vendors
- Enterprise Console provides rules based event correlation capabilities
  - Event Server
    - Caches Events
    - Applies rules to Events
  - Alerts
  - Automatic actions

Tivoli Events

Tivoli Events are Objects (instances of event classes)

Event classes are inherited

Event object contain attributes (called slots)

- Base Event Class contains standard attributes
  - class
  - source (ie LOGFILE, HPOV etc.)
  - hostname
  - status
  - severity
  - acl
  - etc.

Fault Correlation Rules

Each rule stand alone

Rule Format

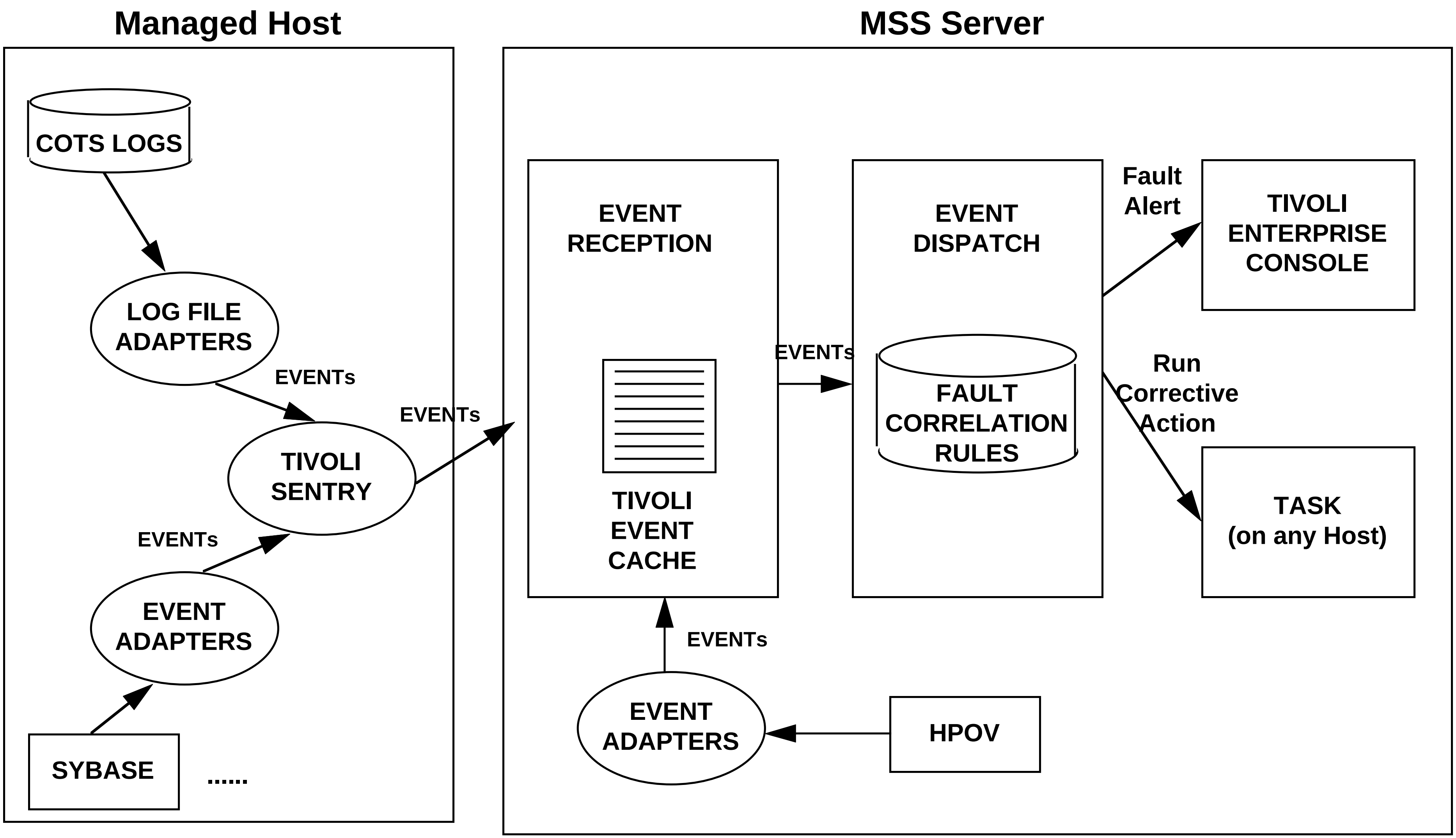
- Rule Type
- Rule Name
- Description
- Filter
- Action

Rules can access all event in event cache

Can execute pre defined primitives

Can define and use variables within scope of rule

Fault Correlation Functional Overview



Trouble Ticketing Overview

- Driving Requirements
- Software Architecture
- Object Model
- Dynamic Model

Trouble Ticketing Driving Requirements

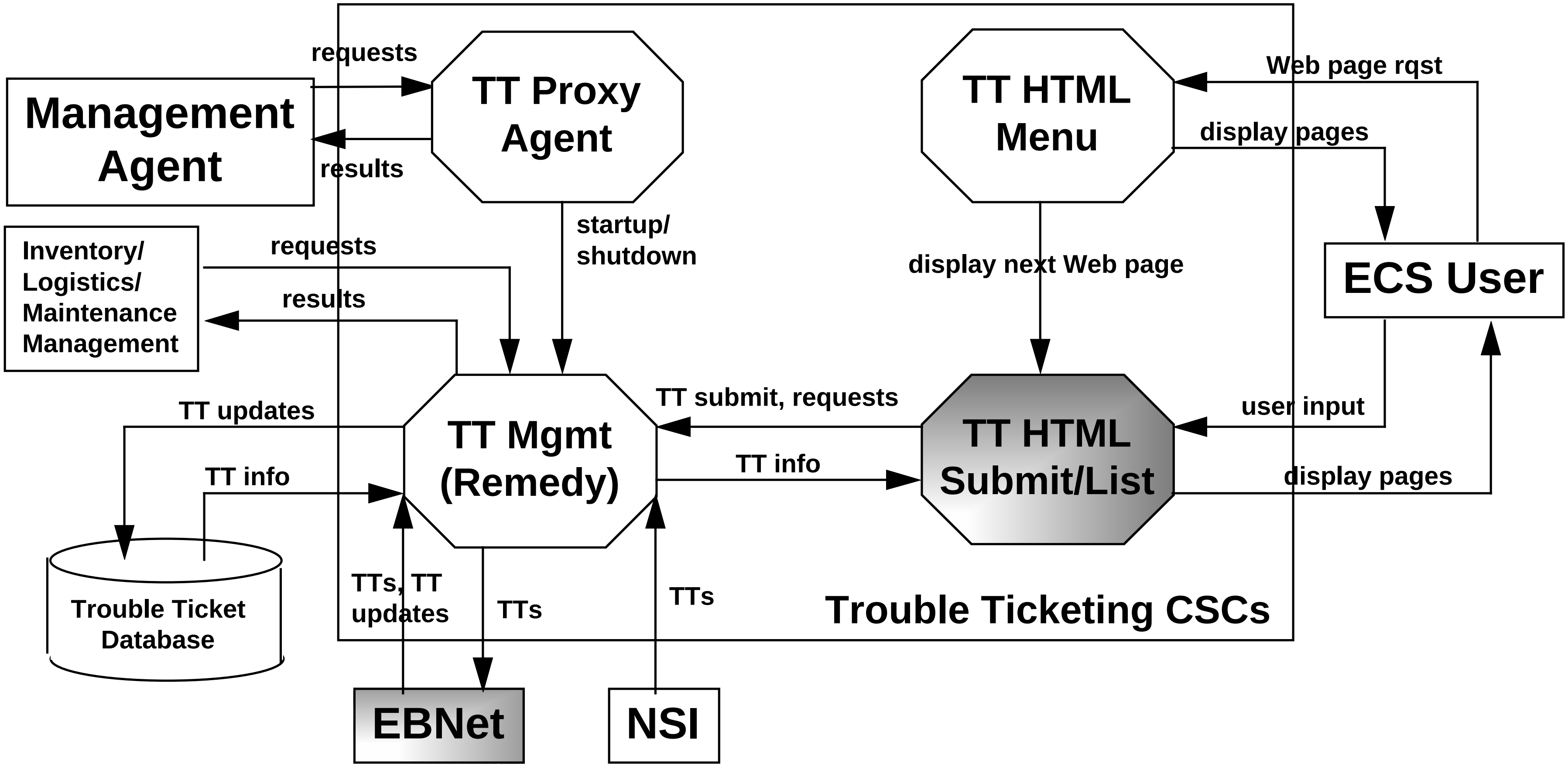
Requirement Summary

- submission of problem reports by ECS users/operators
- maintenance of problem reports by User Service operators

Changes for Release B

- allow submission of Trouble Tickets (TT) if Remedy system is not accessible
- exchange TT and TT status with EBNet

Trouble Ticketing Software Architecture



Trouble Ticketing Object Model Overview

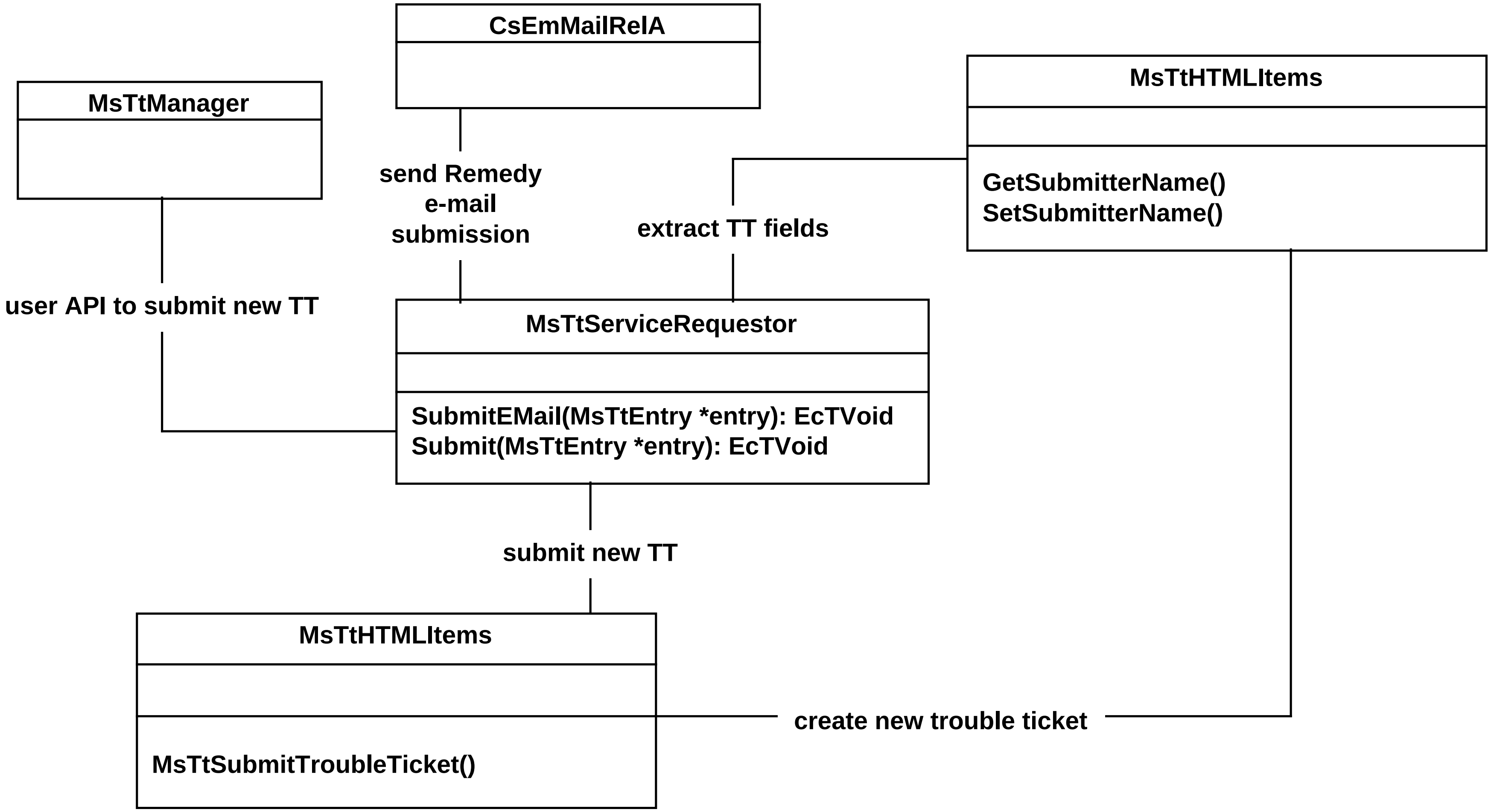
Trouble Ticketing consists of

- Remedy Action Request System (ARS) COTS client/server
  - MsTtManager class
- HTML Interface to submit and list TTs
  - All other classes

For Release B, the following classes change:

- MsTtServiceRequestor class
  - If error is returned from Remedy ARS new TT submittal, formatted e-mail message is created and sent to the Remedy ARS application (via the CsEmMailRelA class) to submit the TT.
- MsHTMLItems class/application
  - If TT submitted via e-mail, confirmation Web page will not show TT number, but will say that TT will be submitted when TT server becomes available and at that time, the user will receive e-mail confirmation including the TT number.
- MsTtManager class (Remedy ARS)
  - Add transfer schema to exchange TTs with EBNet and to receive TT updates from EBNet
  - Add scripts to automatically add and update TTs in ECS schema from transfer schema

Trouble Ticket Object Model - Selected Portions


CsEmMailRelA
MsTtManager
MsTtHTMLItems
GetSubmitterName()
SetSubmitterName()
send Remedy e-mail submission
extract TT fields
user API to submit new TT
MsTtServiceRequestor
SubmitEMail(MsTtEntry *entry): EcTVoid
Submit(MsTtEntry *entry): EcTVoid
submit new TT
MsTtHTMLItems
MsTtSubmitTroubleTicket()
create new trouble ticket

Doc 305-CD-029-002, Section 6.9.3, Trouble Ticket Object Model

Trouble Ticket Dynamic Model

The following event trace will be reviewed (found in the Release B CSMS System Management Subsystem Design Specification for the ECS Project (305-CD-029-002), Section 6.9.4):

- User Submits Trouble Ticket When Remedy is Down (Figure 6.9-4)
  - sequence of events which occur when a registered ECS user submits a TT when Remedy is not accessible

Security Management Overview

- **Driving Requirements**
- **Software Architecture**
- **Object Model**

Security Management Driving Requirements

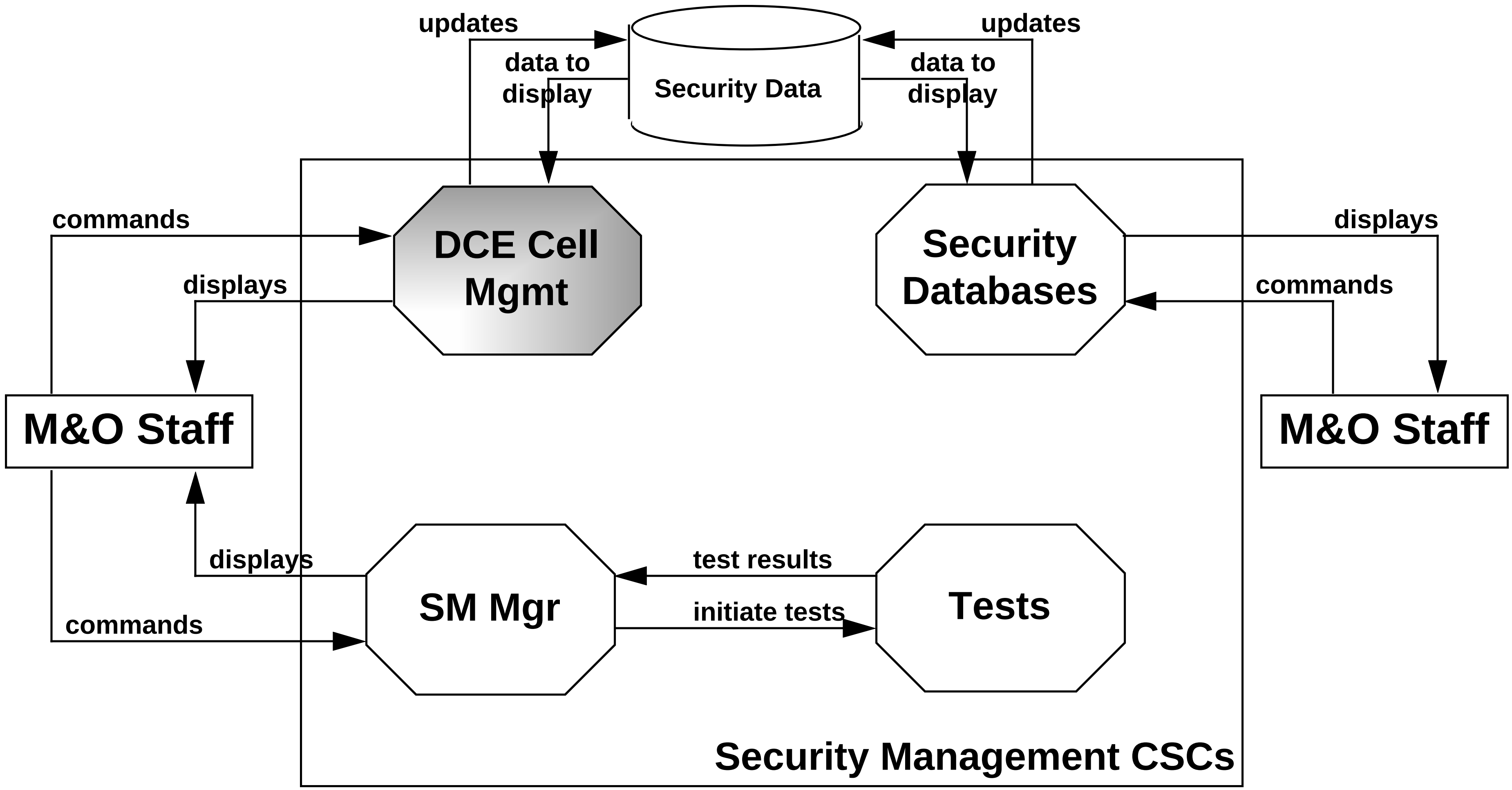
Requirement Summary

- management of security mechanisms which protect and control access to ECS resources

Changes for Release B

- replace COTS product DCE Cell Administration (by HaL) with Hewlett-Packard's acctmgr tool for the purpose of administering the DCE Security Server.

Security Management Software Architecture



Security Management Object Model Overview

Security Management consists of a collection of COTS and public domain tools and some custom developed software to manage the tools

For Release B, the following classes change:

- HPDCEAccntMgr replaces HalDcm for administering the DCE Security Server.
- Proxy agent that managed HalDcm is removed
- EcExtSysIFB and CsEmMailRelA classes are added to provide the interface to external systems for security notices.

Security Management Object Model

The following object model will be reviewed (from the Release B CSMS System Management Subsystem Design Specification for the ECS Project (305-CD-029-002)):

Diagram Name	Document Reference
Security Management Object Model	Section 6.8.3, Figure 6.8-2

User Comment Survey Overview

- Driving Requirements
- Software Architecture
- Data Model
- Object Model
- Dynamic Model

User Comment Survey Driving Requirements

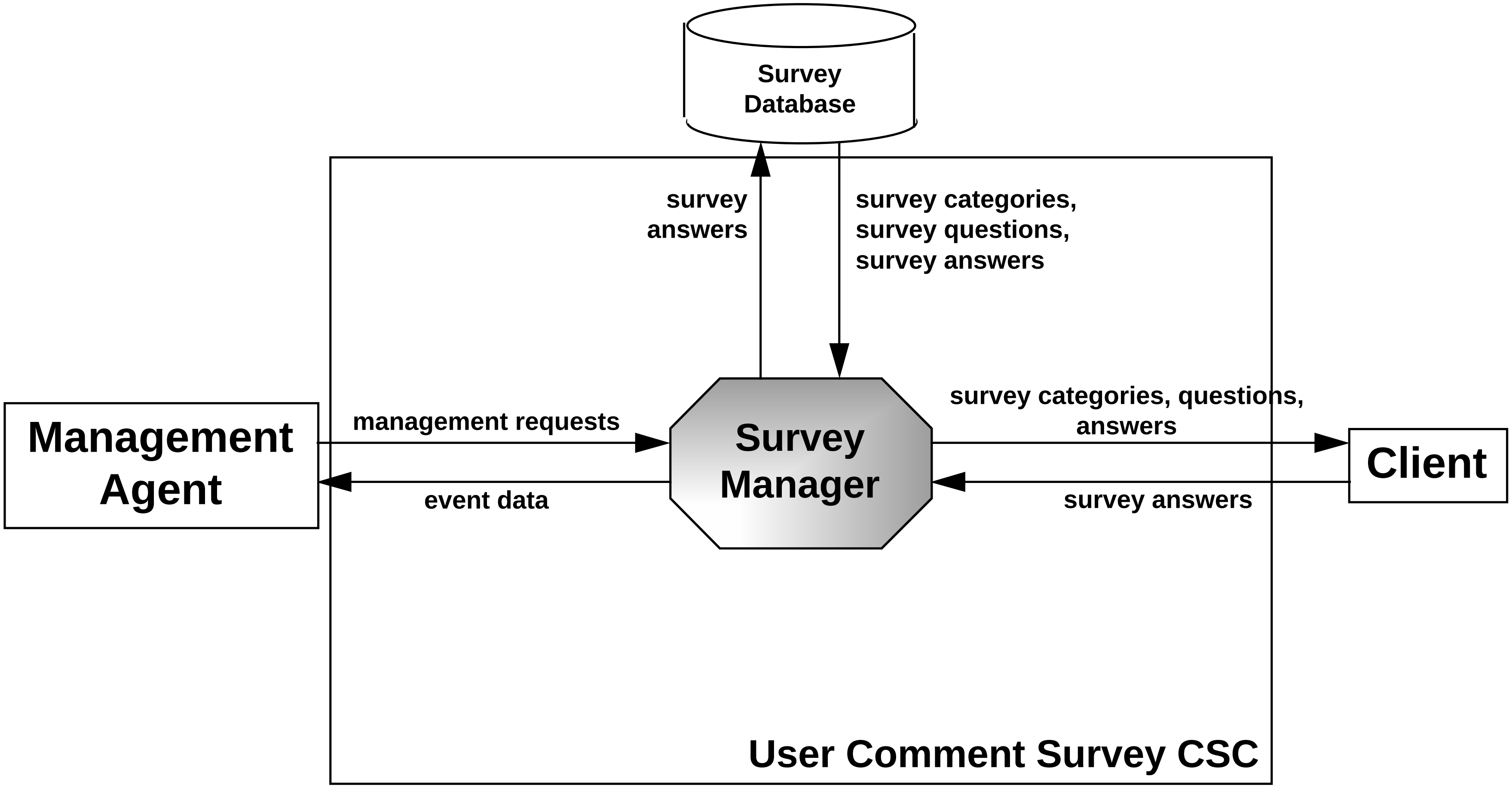
Requirement Summary

- manage/store surveys and survey answers/comments
- provide surveys to Client Subsystem
- receive survey answers/comments from Client Subsystem

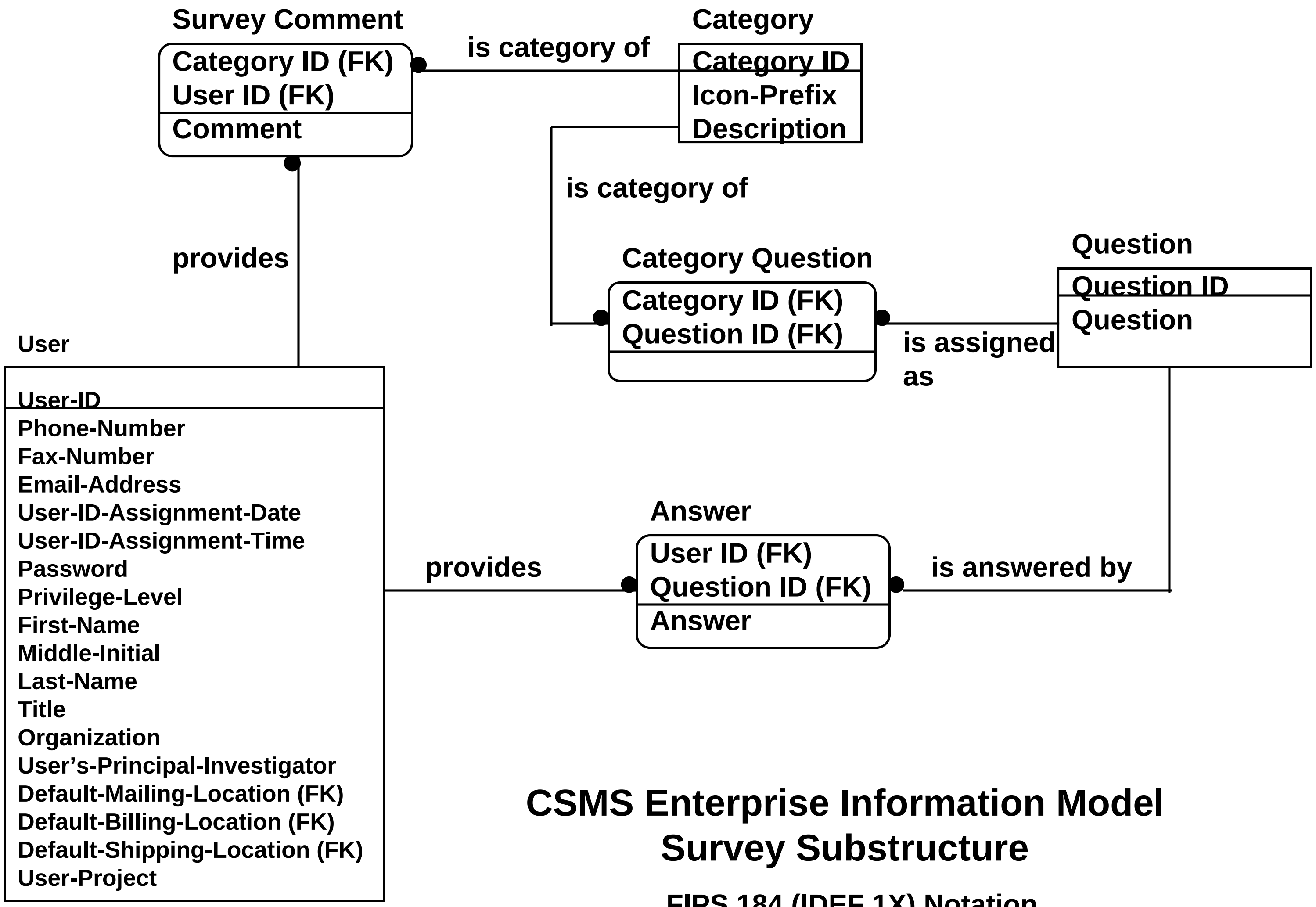
Changes for Release B

- added to provide management of user feedback automatically versus electronic mail.

User Comment Survey Software Architecture



User Comment Survey Data Model



Survey Comment

Category ID (FK)
User ID (FK)
Comment

is category of

Category

Category ID
Icon-Prefix
Description

is category of

provides

Category Question

Category ID (FK)
Question ID (FK)

is assigned as

Question

Question ID
Question

User

User-ID
Phone-Number
Fax-Number
Email-Address
User-ID-Assignment-Date
User-ID-Assignment-Time
Password
Privilege-Level
First-Name
Middle-Initial
Last-Name
Title
Organization
User’s-Principal-Investigator
Default-Mailing-Location (FK)
Default-Billing-Location (FK)
Default-Shipping-Location (FK)
User-Project

provides

Answer

User ID (FK)
Question ID (FK)
Answer

is answered by

CSMS Enterprise Information Model
Survey Substructure

FIPS 184 (IDEF 1X) Notation

User Comment Survey Object Model Overview

User Comment Survey consists of

- MsCsSurveyMgr class - the manager class and the database interface class for the user comment surveys stored in the Sybase Database.
- The other classes are used to collect performance information and to integrate the CSC into the ECS Process Framework.

This custom developed software is new for Release B. In Release A, comments are received via e-mail and processed manually.

User Comment Survey Object Model

The following object model will be reviewed (from the Release B CSMS System Management Subsystem Design Specification for the ECS Project (305-CD-029-002)):

Diagram Name	Document Reference
User Comment Survey Object Model	Section 6.12.3, Figure 6.12-2

User Comment Survey Dynamic Model

The following event trace will be reviewed (found in the Release B CSMS System Management Subsystem Design Specification for the ECS Project (305-CD-029-002), Section 6.12.4):

- **User Fills Out A Survey (Figure 6.12-3)**
  - **sequence of events which occur when a user fills out a Comment Survey.**

Backup Slides

Reference Slides Follow

Fault Correlation Rules - Example

Network File Server named “ROVER” crashes

- Sentry generates "HOST DOWN" event
- Clients to ROVER generate many redundant ‘NFS_SERVER_NOT_RESPONDING’ events

HOST DOWN event is event of concern

- Close other events resulting from ROVER CRASH

Fault Correlation Rules - Example (cont.)

```
rule: close_redundant_events:  (
  event: _thisEvent of class ‘HOST_DOWN’
  where [hostname: _thisHost equals ‘ROVER’ ]
  action: (
      all_instances(
           event: _nfsEvent of class ‘NFS_SERVER_NOT_RESPONDING’
           where [server: equals _thisHost,
              status: _eventStatus within [‘OPEN’, ‘ACK’]]),
      set_event_status( _nfsEvent, ‘CLOSED’)
           )
      ) .
```